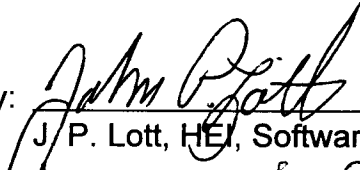

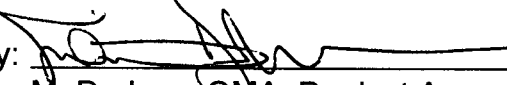
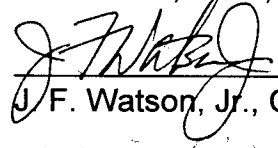

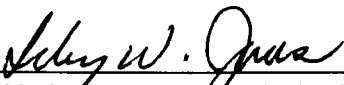
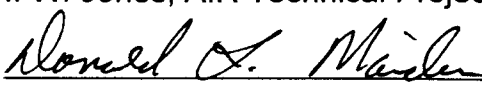


Software Quality Assurance Plan
for the
Atmospheric Ionizing Radiation (AIR) Measurements
Project

March 10, 1997

Submitted By:	 J. P. Lott, HEI, Software Assurance Engineer	<u>3-10-97</u> Date
	 L. J. Johnson, HEI, Sr. Software Assurance Engineer	<u>3-10-97</u> Date
Concurred By:	 M. Dodson, OMA, Product Assurance Engineer	<u>3-24-97</u> Date
	 J. F. Watson, Jr., OMA, Software Assurance Engineer	<u>3-26-97</u> Date
Approved By:	 D. R. Norfolk, AIR Software Manager	<u>3-26-97</u> Date
	 I. W. Jones, AIR Technical Project Engineer	<u>3-26-97</u> Date
	 D. L. Maiden, AIR Program Manager	<u>3-26-97</u> Date

National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia

NAS1-20469

Revision Page

Revision	Date	Description	Approved
--	11-21-96	50% Draft to OMA for review	--
--	01-09-97	Incorporated completed Software Assurance Project Classification Form	--
--	01-28-97	90% Draft to OMA review	02-13-97
--	02-13-97	90% Draft to AIR project review	03-03-97
--	03-10-97	Final Draft issued for approval	03-10-97

Table of Contents

<u>Section</u>	<u>Page</u>
1. INTRODUCTION	1
1.1 IDENTIFICATION	1
1.2 PURPOSE	1
1.3 SCOPE	2
1.4 APPLICABLE DOCUMENTS	2
2. ORGANIZATION AND MANAGEMENT	4
2.1 ORGANIZATION	4
2.2 RESOURCES	4
2.2.1 SQA Equipment, Facilities and Software	4
2.2.2 Customer Furnished Equipment, Facilities and Software	4
2.2.3 Personnel	4
2.3 SCHEDULE	4
2.4 COST	5
3. GENERAL SOFTWARE QUALITY ASSURANCE METHODS	6
3.1 AUDITS	6
3.1.1 Audit Planning	6
3.1.2 Audit Areas	6
3.1.3 Audit Execution	7
3.2 EVALUATING PROCESSES AND PROCEDURES	7
3.3 REPORTING AND CONTROL	8
3.3.1 Problem Reporting	8
3.3.2 Corrective Action	9
3.3.3 Quality Assurance Records	10
3.4 SURVEYS	11
3.4.1 Spot Checks / Inspections	11
3.4.2 Code Analysis / Walkthroughs	11
4. PROJECT LIFE CYCLE PHASE SOFTWARE QUALITY ASSURANCE ACTIVITIES	12
4.1 SOFTWARE CONCEPT AND INITIATION PHASE	12
4.2 SOFTWARE REQUIREMENTS PHASE	12
4.3 SOFTWARE DESIGN PHASE	13
4.3.1 Project Documentation Review	13
4.4 SOFTWARE IMPLEMENTATION PHASE	13
4.5 SOFTWARE INTEGRATION AND TEST PHASE	14
4.5.1 Testing	14
4.5.1.1 Software Test Plan	14
4.5.1.2 Software Test Specifications and Procedures	14

Table of Contents (continued)

<u>Section</u>	<u>Page</u>
4.5.1.3 Software Test Performance.....	15
4.5.1.4 Software Test Assurance Reports.....	15
4.5.1.5 Installation and Checkout.....	15
4.5.2 Verification and Validation	15
4.6 SOFTWARE ACCEPTANCE AND DELIVERY PHASE	16
4.7 SOFTWARE SUSTAINING ENGINEERING AND OPERATIONS PHASE.....	17
4.7.1 Delivery and Installation.....	17
4.7.2 Maintenance	17
5. PROVIDER / SUBCONTRACTOR CONTROL.....	17
6. ACRONYMS / GLOSSARY	18
7. CHECKLISTS / TOOLS.....	19

APPENDICES

Appendix A	OSEMA and AIR Team Organization Charts
Appendix B	AIR Software Milestone Schedule and SQA activities
Appendix C	System/Software Problem Report (SPR)
Appendix D	AIR Software Quality Assurance Checklists

1. INTRODUCTION

This Software Quality Assurance Plan describes the organization and procedures to be used by the National Aeronautics and Space Administration (NASA), Langley Research Center (LaRC) to ensure that the software to be delivered for the Atmospheric Ionizing Radiation (AIR) project complies with the application requirements, as well as the technical standards set by the NMI 2410.10B NASA Software Management, Assurance, and Engineering Policy, the NASA-STD-2201-93 Software Assurance Standard, and the NASA-STD-2100-91 Software Documentation Standards.

1.1 Identification

This document is identified as the NASA LaRC Software Quality Assurance Plan (SQAP) for the AIR project. This plan establishes the policies, standards, procedures, and practices for software assurance of all computer software for applications developed for or by LaRC for the AIR project. Moreover, this plan establishes a standards compliance perspective for LaRC.

The AIR Environmental Reconnaissance (ER-2) measurements project is part of a program to study the radiation risk to the high-altitude flight operation of a commercial supersonic transport. Because the complexity of the radiation environment at high altitudes requires a diverse array of instruments, the AIR payload will consist of 13 sensor instruments, with the dominant instrument being a set of neutron radiation detectors (Bonner Spheres) from the Department of Energy's Environmental Measurements Laboratory (EML). The AIR payload will be installed in the ER-2 aircraft for a series of flights from three operating locations (Ames, Fairchild and Wallops) beginning in 1997.

The AIR payload is designed and developed by LaRC to integrate the instruments into racks that fit the ER-2's Nose, Q-bay, and Right and Left Superpods. The AIR payload includes 13 sensor instruments and a data acquisition system that will: record data from the EML Bonner Spheres and Scintillation Counters, record aircraft position and system health monitoring data, and provide instrument status to the pilot's control panel and a data stream to the ER-2 telemetry system. The data acquisition computer will be located in the right superpod. The payload will be removed and stored until the subsequent series of flights are scheduled for the AIR project.

1.2 Purpose

The purpose of the SQAP is to ensure the quality of newly developed and modified software. This shall be accomplished by the following software assurance tasks: Audits, Reporting and Control, Life Cycle Reviews, and Surveys which include Process Evaluations, Spot checks / Inspections, and Code Analysis / Walkthroughs.

The LaRC Office of Mission Assurance (OMA) has the responsibility for assuring that all software development phases of the software life cycle are conducted in accordance with requirements and standards as noted in Section 1.4.

1.3 Scope

The scope of this document is to define the activities which shall be carried out in support of software quality assurance for the AIR project.

This document defines the actions that provide assurance that the software-related items delivered conform to their established and contracted technical requirements. Software Quality Assurance (SQA) will also ensure that standards, practices, procedures, and methods of the software development process are defined and applied.

This plan contains 1) orientation material, 2) the organization and resources that shall be used to accomplish the SQA activities, 3) the planned schedule of SQA activities, 4) the planned SQA activities and the tools/procedures that support those activities, and 5) the quality assurance records that shall be kept.

This plan conforms to the requirements of NASA-STD-2201-93, ISO 9000-3, and SEI-TR-25.

1.4 Applicable Documents

The documents shown below, form a part of this document to the extent described herein. In the event of a conflict between the documents referenced herein and the contents of this document, the contents of this document shall be considered a superseding requirement to the previous documents.

The following list contains all related documents which provide supporting information pertinent to the information in this SQAP:

1. AIR Mission Plan
2. AIR Product Assurance Plan (AIR-4.0.1)
3. AIR Configuration Management Plan (AIR-XXX) - TBD

Other plans related to the quality of software development may include the Software Management/Development Guidelines (SM/DG) which describes the standards, schedule and procedures to be followed by the development team, the Software Configuration Management Plan (SCMP) which describes how products of this project shall be identified, baselined and controlled, and the Software Test Plan and Procedures (STPP) which describes how products of this project shall be tested.

This plan conforms to the following requirements and standards:

ISO 9000-3:1991	Quality management and quality assurance standards -- Part 3; Guidelines for the application of International Organization for Standardization (ISO) 9001 to the development, supply and maintenance of software.
SEI-TR-25	Key Practices of the Capability Maturity Model (CMM), Version 1.1, Carnegie Mellon University (CMU), Software Engineering Institute (SEI), February 1993.
NASA-STD-2201-93	NASA Technical Standards Division, Software Assurance Standard, November 10, 1992
LHB 5300.4	Software Quality Assurance Handbook, October 1995
LMI 7120.1	Flight Projects and Experiments Review Program, June 27, 1995
LMI 7120.2	Authority and Responsibilities of Managers of Small Space Flight Projects, April 5, 1990
LMI 7122.1	Langley Research Center Systems Engineering Process for In-House Space Flight Projects, August 5, 1994

2. ORGANIZATION AND MANAGEMENT

2.1 Organization

The OMA organization is a separate organization (i.e., separate from the engineering organization) which reports directly to the Project Manager (PM). The organization is obligated to fully advise the PM of the status of the software and the accompanying documentation.

The OMA is responsible for the fulfillment of, and for ensuring compliance with, the software quality assurance activities defined in this document. The OMA has clearly defined authority and responsibility. The OMA provides senior management with an independent evaluation of the effort, and it provides a direct reporting line to the senior management of the AIR project to resolve problems.

2.2 Resources

2.2.1 SQA Equipment, Facilities and Software

The OMA shall use a personal computer with applicable software tools for creating reports and documents, tracking SQA activities, and other activities defined in this plan.

2.2.2 Customer Furnished Equipment, Facilities and Software

There are no requirements for using customer furnished facilities in executing this SQAP.

2.2.3 Personnel

The OMA shall select SQA representatives experienced with the activities of the Software Life Cycle as defined by LHB 5300.4. There shall be one SQA representative for the AIR project. The SQA representative is responsible for evaluating the quality (usability, reliability, etc.) of the software being developed. They have background in Software Engineering principles with experience in developing software.

Appendix A shows the Office of Safety, Environment and Mission Assurance (OSEMA) roles and responsibilities chart as it pertains to the working relationship when software assurance is involved in a LaRC software developmental project.

2.3 Schedule

Appendix B presents the major milestones of the software development schedule for the AIR project and the corresponding SQA activities that shall be performed during this development. The SQA activities shall be performed continuously during the AIR

project. Since the OMA SQA representative was not involved with the AIR project until after the CDR, the SQA activities shall include reviewing all previous documents, reviews, etc. and begin the SQA activities with the Software Integration and Test Phase of the AIR project.

2.4 Cost

The software quality assurance activities for the AIR project shall be processed by the OMA.

3. GENERAL SOFTWARE QUALITY ASSURANCE METHODS

In accordance with LaRC procedures described in the LHB 5300.4, the OMA has classified the software for the AIR project as LOW. The software assurance activities shall be performed by the OMA SQA representative assigned to the AIR project.

3.1 Audits

The OMA shall audit the software development and related processes for the AIR project according to the procedures described in this section.

Throughout the life cycle of the project, the OMA shall perform audits to determine whether software procedures, standards, and practices have been identified and are being properly implemented. Audits also identify those areas in which additional controls and standards are required to ensure the quality of the software product. An audit may be a stand-alone activity or incorporated into a review. All audit reports will be forwarded to the Director of OSEMA.

3.1.1 Audit Planning

The development and maintenance of schedules for internal audits shall be the responsibility of the OMA. These schedules shall be based on software life cycle phases, specific software products of each phase, and previous audits results.

Prior to the start of the audit, the following activities shall be completed:

- Review of the standards, practices, requirements, approved procedures, and directives of the area or function to be audited,
- Preparation of checklists as needed, and
- Establishment of the audit team and the functions, procedures, and data to be audited.

3.1.2 Audit Areas

Audits by OMA shall be conducted on the following areas:

- System Definition
- Software Design and Development
- Configuration Management
- Software System Test
- Sustaining Engineering Support

The OMA SQA representative shall use the checklists in Section 7 of this plan for the assessment of SQA activities for the AIR project.

3.1.3 Audit Execution

The SQA representative may request support from the engineering organization responsible for the area to be audited. This support may be in the form of personnel, records, files, and procedures. Examinations during an audit may include both processes and products. The results and findings of an audit shall be documented in a report following the guidelines outlined in Section 3.3 of this plan.

3.2 Evaluating Processes and Procedures

The following software processes, as noted in the LHB 5300.4, will be evaluated by an assessment panel selected by OMA and chaired by the OMA Software Assurance Engineer.

- Requirement Management -- *will not be evaluated for AIR*
 - System requirements allocated to software are controlled to establish a baseline for software engineering and management use.
 - Software plans, products, and activities are kept consistent with the system requirements allocated to software.
- Software Project Planning -- *will not be evaluated for AIR*
 - Software estimates are documented for use in planning and tracking the software project.
 - Software project activities and commitments are planned and documented.
 - Affected groups and individuals agree to their commitments related to the software project.
- Software Project Tracking and Oversight -- *will not be evaluated for AIR*
 - Actual results and performances are tracked against the software plans.
 - Corrective actions are taken and managed to closure when actual results and performance deviate significantly from the software plans.
 - Changes to software commitments are agreed to by the affected groups and individuals.
- Software Product Engineering
 - The software engineering tasks are defined, integrated, and consistently performed to produce the software.
 - Software work products are kept consistent with each other.
- Software Quality Assurance
 - Software quality assurance activities are planned.

- Adherence of software products and activities to the applicable standards, procedures, and requirements is verified objectively.
- Affected groups and individuals are informed of software quality assurance activities and results.
- Noncompliance issues that cannot be resolved within the software project are addressed by senior management.
- Software Configuration Management
 - Software configuration management activities are planned.
 - Selected software work products are identified, controlled, and available.
 - Changes to identified software work products are controlled.
 - Affected groups and individuals are informed of the status and content of software baselines.
- Software Contract Management -- *will not be evaluated for AIR*
 - The prime contractor selects qualified software subcontractors.
 - The prime contractor and the software subcontractor agree to their commitments to each other.
 - The prime contractor and the software subcontractor maintain ongoing communications.
 - The prime contractor tracks the software subcontractor's actual results and performance against its commitments.

3.3 Reporting and Control

3.3.1 Problem Reporting

When an anomalous condition is found in the development project life cycle or maintenance process, the SQA representative shall document the problem on the System/Software Problem Report (SPR) or applicable document used by the software development organization. Appendix C of this plan contains a copy of the SPR form with instructions. These reports are the end result of reviews, audits, and tests. The SPR will be tracked from its inception to resolution by the SQA representative.

The SQA representative has the responsibility of establishing and maintaining a project SPR file. Each file shall contain the SPR, inclusive of checklists, notes and procedures, the corrective action replies, and any associated correspondence.

During reviews, audits, and testing, a SPR shall be written against questionable items. These SPRs shall be answered by the appropriate engineering organization within a 30-day period. The OMA has the responsibility to monitor the SPR process to ensure that all SPRs are answered correctly and completely.

3.3.2 Corrective Action

The SQA representative shall evaluate the corrective action system for handling SPRs according to the procedures described in this section. The objective of the corrective action system is to provide a systematic method of resolving identified problems. Identification of problems may be based on:

- a. contractual, company, departmental, or program procedural nonconformance;
- b. unsatisfactory quality trends discovered in internal reviews and walkthroughs;
- c. functional deficiencies and performance problems traceable to software errors, operator errors, testing, or inspection discrepancies;
- d. unsatisfactory supplier or subcontractor performance; and
- e. inconsistencies, errors, and omissions in specifications and documents.

Each time a software failure is observed or a software error is discovered, a SPR form shall be completed to make certain that corrective actions are taken to remove the error.

Action items requiring corrective action may result from reviews and walkthroughs. When corrective action is taken and verified by the review/walkthrough leader, the completed action item list is sent to the OMA. Regardless of the source of the problem, the SQA representative shall conduct a follow-up inspection of the corrected documents or code to verify correction of the original deficiency.

The following characteristics are essential of the corrective action system:

- a. correct, complete, and accurate problem reporting;
- b. careful analysis of the problems reported;
- c. proper classification of the problems as to category and severity;
- d. assignment of responsibility for correcting the problem;
- e. determination of effective actions to be taken to remedy the problem;
- f. regular, thorough analysis of trends;
- g. effective, implementable recommendations resulting from trend analysis;
- h. proper authorization of steps to correct problems;
- i. complete records of documented actions taken;
- j. re-evaluation of corrective actions after being taken;
- k. tracking corrective action progress and closing out completed actions; and
- l. customer visibility into the corrective action process.

Section 7 contains a checklist for the SQA evaluation of the operation and effectiveness of the software development organization's corrective action system.

3.3.3 Quality Assurance Records

Records of the SQA activities described in this plan will be kept in the OMA AIR files. These records may be inspected at any time.

The OMA AIR files may include the following records:

- Annotated copies of AIR documents that have been the subject of SQA Inspections, Audits and Reviews
- Data and Analysis Graphics for Software Metrics
- Completed Checklists
- Internal Review Reports
- Approval Record Sheets
- Monthly SQA Reports
- Document Review Reports
- Test Observation Records
- Problem Reports and Corrective Action Records
- Issue Tracking Forms

To assist in making and evaluating the software development products and processes, basic software metrics shall be used. These basic metrics are:

- 1) Product size, of the total software product and of each major component of that product.
- 2) Person-months for the total project and by project phase and for major software components.
- 3) Schedule time for the total project and by project phase.
- 4) Number of software errors made and discovered in each project development phase.
- 5) Number of software errors discovered after completion of the software development.
- 6) Software requirements volatility (i.e., the percent of software errors that have been modified or added after the end of the software requirements definition phase).
- 7) Number of tests successfully executed as a percentage of the total number of tests planned.
- 8) The total number software problem reports written compared with the number of software problem reports that have been closed.

The SQA representative shall support the collection of data for these metrics and shall prepare reports documenting the conclusions of the analysis of the metrics data. Graphic plots of the above metric values versus time shall be used by the SQA

representative to reveal trends. The SQA representative shall also contrast the metric values obtained during the AIR project with values obtained from past projects.

3.4 Surveys

3.4.1 Spot Checks / Inspections

The content of this section, as described in LHB 5300.4, is not applicable to the AIR project.

3.4.2 Code Analysis / Walkthroughs

The content of this section, as described in LHB 5300.4, is not applicable to the AIR project.

4. PROJECT LIFE CYCLE PHASE SOFTWARE QUALITY ASSURANCE ACTIVITIES

Life Cycle reviews are formalized activities which maintain visibility into the process of software development. These reviews are scheduled in the beginning of the project and represent the necessary milestones in the development process. The responsible organizations initiating the review shall ensure proper notification of impending reviews. This section contains detailed SQA activities for each life cycle review phase.

The status and results of SQA activities within a given life cycle phase shall be presented based on Project Manager concurrence.

In conducting the AIR life cycle phase software quality assurance activities, the SQA representative shall use the checklists provided in Section 7 of this plan.

4.1 Software Concept and Initiation Phase

The SQA representative shall review the AIR SM/DG to verify that it complies with all requirements and describes a practical and effective software development process. In particular, the SQA representative shall confirm that adequate provisions have been made for error prevention and early error discovery procedures, techniques and tools.

The SQA representative shall review the allocation of user needs and system requirements of the product and process items to confirm that no requirement has remained unallocated.

4.2 Software Requirements Phase

The SQA representative shall review the software requirements to make certain they are complete and consistent and are a suitable foundation for the subsequent design and test activities. In analyzing the quality of requirements, special attention will be paid to the "testability" of each requirement. That is, if there is no straightforward way to test or inspect whether or not a requirement is met, then the statement of the requirement is deficient.

The SQA representative shall review the AIR Software Requirements document produced by the software development organization. This document shall be checked for compliance with its applicable document preparation standard to verify it has the requisite format and content.

4.3 Software Design Phase

The SQA representative shall review the software design documents. These documents shall be examined to make certain they are in their required form and have the required content.

4.3.1 Project Documentation Review

Documentation throughout life cycle of the project must be inspected and checked to ensure that it is correct. The following criteria shall be used:

- Compliance to documentation requirements and standards
- All necessary information has been included
- Text is clear and unambiguous
- Adequate traceability and cross-referencing has been provided and is maintained

Each phase of the life cycle generates different documents which need to be inspected. These documents shall be subjected to a SQA inspection prior to the formal review of which it will be a part.

The following is a list of documentation:

- Project Management Plan
- Functional Requirements Document
- Software Management / Development Guidelines
- Software Design Specification
- User and Operations Guide
- Training Plan and Procedures
- Integrated Test Plan
- Software Test Plan and Procedures
- Site Installation Plan

4.4 Software Implementation Phase

The source code is the programming language implementation of the software design. The SQA representative shall confirm that the implemented code is traceable to the functions specified in the design documents produced in earlier project phases and shall also confirm that the source code follows coding standards. Prior to the software development organization submitting the code for software integration, the software code is to be subjected to SQA for review.

4.5 Software Integration and Test Phase

The SQA representative shall review the software test plan to ensure that it describes a software test activity that thoroughly exercises the software implementation and provides evidence that all software requirements have been satisfied.

The SQA representative shall review the software test procedures to ensure that they are consistent with the software test plan and describe effective and comprehensive tests. A major criterion in evaluating these test procedures is to make certain that the descriptions of the test are sufficient for a person other than the test procedure author to conduct the test.

The SQA representative shall witness selected software tests and review the results of many additional tests that it does not witness. The purpose of this test witnessing and the test results review is to confirm that the tests were run as described in the test procedures and that the results are acceptable.

4.5.1 Testing

Testing is an activity performed by the software development organization and reviewed by the SQA representative. Its primary concern is the controlled exercise of the program code using sample input cases with the objective of exposing errors. This process begins with the smallest unit of the system, continues up to, and includes installation.

In order to accomplish this task, it is necessary to generate a Software Test Plan and Procedures (STPP) document. This product must be reviewed by the responsible engineering organization manager and SQA representative to ensure their completeness according to standards.

4.5.1.1 Software Test Plan

The test plan shall contain test objectives, methodologies (from unit test to installation and checkout), description of the test environment, test description, delineation of the requirements verified, an evaluation plan, and a test schedule.

The test plan is begun in the SRR and revised as the system matures. This maturation process continues until CDR. Prior to PDR and again prior to CDR this plan is submitted for the SQA representative to review.

4.5.1.2 Software Test Specifications and Procedures

Software test specifications and procedures are step-by-step descriptions of the actions necessary to execute a particular test. Each test procedure shall contain the scope of the test, the applicable documents, organizational responsibilities, test environment,

test data, criteria for acceptance, and the steps necessary to complete the test. Each individual step should be comprised of the user input and the expected result. All test procedures shall be subjected to a SQA review prior to the test. The SQA representative shall review the procedures to determine their completeness based on requirements and contractual agreement.

4.5.1.3 Software Test Performance

The SQA representative shall be notified of all software related tests. It is the responsibility of the SQA representative to note any deviation from the accepted procedures. As the test is being conducted, the software development organization shall initial each step as it is completed. Hard copies of data (logs or copies of screen images) and output media (diskette or tape) shall be baselined by the software development organization. The SQA representative shall review these test results and related documents.

4.5.1.4 Software Test Assurance Reports

Upon completion of the test, a post test briefing shall be conducted by the test team. This will occur within a one week period in order to give time for any off-line analysis that must take place. During this meeting with the test personnel, programmers, customer, and the SQA representative, the results of the test shall be reviewed. Any problems noted during the test and any problems found in analysis shall be documented with a SPR form. If there is any question regarding the severity of the problem, the SQA representative with the Project Managers concurrence shall be the final judge. The results of the post test briefing shall be written by the test team and reviewed by the SQA representative.

4.5.1.5 Installation and Checkout

Software which passes testing is ready for installation at the customer's site. The SQA representative shall ensure that all schedules for new or modified hardware have been met and that new or modified hardware has been tested and is ready to accept the tested software. This will be a cold start situation. After the system has been brought up, the tested software is loaded and a rerun of the final integrated test is conducted. The results of this rerun shall be compared to the original test to determine if the system is ready for operation.

4.5.2 Verification and Validation

Software verification and validation is the process of ensuring that software being developed or changed will satisfy functional and other requirements (validation) and each step in the process of developing the software is accomplished (verification). The SQA representative shall validate test procedures to specified requirements, verify the

test plan implementation, witness performance tests, and have the capability to modify the test plan to include any special customer requested testing.

4.6 Software Acceptance and Delivery Phase

Prior to the final release of a deliverable (software or document), it must meet the approval of the AIR Project Manager and the leader of the AIR SQA representative. The necessary approval is signified by including a block of signatures on the cover sheet of the Release document, or on the software delivery cover letter.

The procedure leading up to the sign-off on the deliverable by the leader of the AIR SQA representative, includes having passed the designated SQA milestones specified in the life-cycle steps. In addition to the basic approval steps, the leader of the AIR SQA representative performs his own final check of the deliverable. These steps are documented below:

- (1) Verify that all SQA documents or code were successfully completed, and that required signatures have been obtained and Verification Description Document is accurate.
- (2) Assess the status of outstanding problems to determine if they are all closed and that corrective actions have been implemented and will be completed before acceptance. Deliverables may be released with outstanding problems only in an emergency.
- (3) Perform an independent assessment of the deliverable (read the document or exercise the software) to form an opinion as to the quality level and the image which will result from the release of the deliverable.
- (4) If a hold on the deliverable is warranted, meet with the AIR Project Manager to discuss the criticality of the timing of the release and the quality of the product to be released. Decide, based upon this discussion, whether or not to sign off on the delivery.
- (5) If an impasse is reached in the decision whether or not to authorize the delivery, document this fact as an Issue, and notify the next higher level of management.
- (6) If the decision by the next higher level of management is to make the delivery, the delivery sign off on the "SQA Manager" line of the cover sheet shall be co-signed by the manager making the decision. In any case, the OMA may continue to appeal this decision to successively higher management organizations.

4.7 Software Sustaining Engineering and Operations Phase

4.7.1 Delivery and Installation

The SQA representative shall confirm that all copies are identical to the copy approved as a part of the Acceptance Phase. The SQA representative shall verify that the installation instructions that accompany the delivery are correct and complete.

4.7.2 Maintenance

The SQA representative shall continue to monitor the maintenance of the AIR software products after delivery to make certain that the quality and reliability do not degrade as a result of maintenance actions. In monitoring the maintenance actions, the SQA representative shall follow the procedures described in Section 3 and 4 of this plan. The SQA records shall continue to be maintained and updated during the SQA support of the maintenance activity. During the maintenance activity particular attention will be paid to the continuing operation of the problem reporting and corrective action system and making certain the software configuration management procedures are still followed.

5. PROVIDER / SUBCONTRACTOR CONTROL

The content of this section, as described in LHB 5300.4, is not applicable to the AIR project.

6. ACRONYMS / GLOSSARY

This section contains an alphabetical list and definitions of all acronyms and abbreviations used in the document, and any word used in a non-standard way.

AIR	Atmospheric Ionizing Radiation
CDR	Critical Design Review
CMM	Capability Maturity Model
CMU	Carnegie Mellon University
EML	Environmental Measurements Laboratory
ER-2	Environmental Reconnaissance-2
ISO	International Organization for Standardization
KSC	Kennedy Space Center
LaRC	Langley Research Center
LHB	Langley Handbook
LMI	Langley Management Instruction
NASA	National Aeronautics and Space Administration
NMI	NASA Management Instruction
OMA	Office of Mission Assurance
OSEMA	Office of Safety, Environment and Mission Assurance
PDR	Preliminary Design Review
PM	Program Manager
QA	Quality Assurance
SCMP	Software Configuration Management Plan
SEI	Software Engineering Institute
SM/DG	Software Management / Development Guidelines
SPR	System/Software Problem Report
SQA	Software Quality Assurance
SQAP	Software Quality Assurance Plan
STPP	Software Test Plan and Procedures

7. CHECKLISTS / TOOLS

Appendix D shows the checklists from the LHB 5300.4 that shall be used by the OMA SQA representative when evaluating the software assurance activities for the AIR project. No automated code analysis tools are utilized for the AIR project.

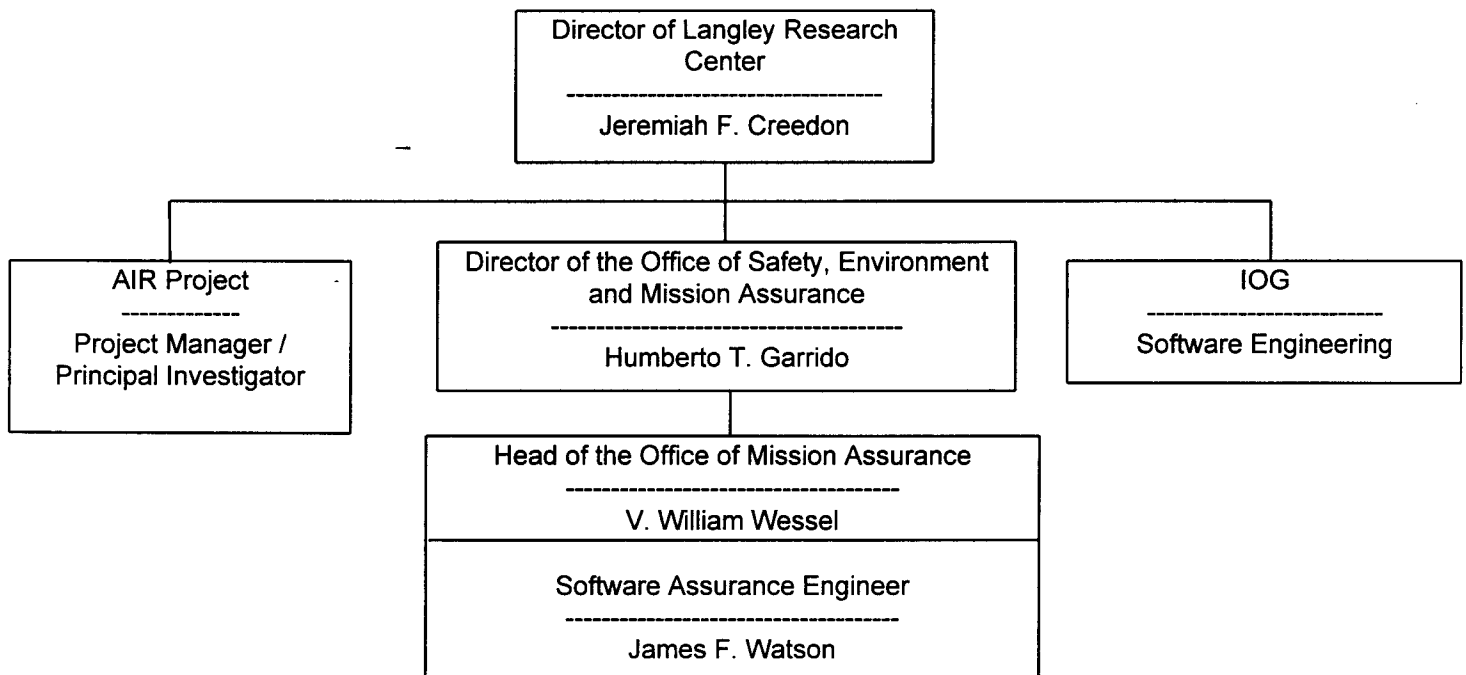
<u>Table</u>	<u>Title</u>	<u>Page</u>
1	Software Quality Evaluation Checklist	D-2,3
2	Software Configuration Management Review Checklist	D-4,5
3	Software_Development Library (SDL) Review Checklist	D-6,7
4	Corrective Action System Evaluation Checklist	D-8
5	Hardware - Software Allocation Checklist	N/A
6	Software Requirements Document Review Checklist	D-9
7	Software Design Document Review Checklist (may have some impact on the Test Plan)	D-10,11,12
8	Software Code Review Checklist	N/A
9	Software Test Plan Review Checklist	D-13,14
10	Software Test Procedure Review Checklist	D-15
11	Software Test Conduct Review Checklist	D-16

N/A -- Not applicable to the AIR project

Appendix A

Office of Safety, Environment and Mission Assurance (OSEMA)

Organization Chart



AIR -- Team Organization Chart

The AIR Team Organization Chart is maintained by the Project Manager.

Appendix B

AIR Software Milestone Schedule

Milestones:

The Schedule and Resource Analyst maintains the complete schedule for the AIR project.

SQA Activities:

Develop Software Quality Assurance Plan

Evaluating Processes and Procedures
Reporting and Control

Software Concept and Initiation Phase

-- Review the standards

Software Requirements Phase

-- Review Software Requirements document

Software Design Phase

-- Review all AIR documentation

Software Implementation Phase

-- Ensure traceability of the software code

Software Integration and Test Phase

-- Review STPP and unit testing procedures

-- Review / Witness software testing

Software Acceptance and Delivery Phase

-- Final check (items 1-6 Section 4.6)

Software Sustaining Engineering and Operations Phase

-- Monitor the maintenance

-- Maintain and update records

Appendix C

System / Software Problem Report (SPR)

The following is the System/Software Problem Report (SPR) form. It shall be completed by the SQA representative when an anomalous condition is found in the development or maintenance process.

Instructions:

Part A, with the exception of number 3 (furnished by the SQA representative), is to be completed by the individual that found the problem. Be as specific as possible.

Part B shall be completed by the cognizant engineer, the Project Manager, or Software Development lead.

Part C shall be completed by the Configuration Control Board (CCB) for the AIR project.

Parts D and E will be signed when the corrective action has been completed.

Any questions regarding this form should be addressed to the SQA representative for the AIR project.

SYSTEM/SOFTWARE PROBLEM REPORT (SPR)

PART A - ORIGINATOR

1. PROJECT:		2. DATE:		3. SPR NUMBER:	
4. NAME: (Print)		5. MAIL STOP:	6. PHONE:		7. ORGANIZATION:
8. FREQUENCY: <input type="checkbox"/> EVERY TIME <input type="checkbox"/> INTERMITTENT <input type="checkbox"/> ONCE	9. CAN THE PROBLEM BE REPRODUCED AT WILL? <input type="checkbox"/> YES <input type="checkbox"/> NO	10. TYPE OF REPORT: <input type="checkbox"/> PROBLEM REPORT <input type="checkbox"/> SUGGESTED ENHANCEMENT <input type="checkbox"/> DOCUMENTATION ERROR		11. APPLICATION IMPACT: <input type="checkbox"/> FATAL <input type="checkbox"/> MODERATE <input type="checkbox"/> SERIOUS <input type="checkbox"/> ANNOYING	
12. DESCRIPTION OF PROBLEM:					
13. SEQUENCE OF EVENTS:					
14. ERROR MESSAGE(s)					

PART B - COGNIZANT ENGINEER

15. COGNIZANT ENGINEER:	DATE:	ORGANIZATION:
16. ANALYSIS OF ERROR:		
17. DOCUMENTS AND CODE INFECTED AND AFFECTED BY THE PROBLEM:		
18. CHANGES MADE OR ACTIONS TAKEN TO CORRECT THE PROBLEM:		

PART C - CONFIGURATION CONTROL BOARD (CCB)

19. RECOMMENDATIONS:	STATUS:	DATE:
----------------------	---------	-------

PART D - AUTHORIZATION

20.	Cognizant Engineer	Date	Software Assurance Representative	Date
	Project Manager or Software Lead	Date	Other	Date

PART E - VERIFICATION AND CLOSE OUT

21. INITIAL FOR COMPLETED ACTIONS				22. CLOSE OUT AUTHORIZATION:	
a. Corrections implemented: BY:	b. Corrections inspected: BY:	c. Test/retest verification: BY:	d. Documentation verification: BY:	BY:	BY:
DATE:	DATE:	DATE:	DATE:	DATE:	DATE:

Appendix D

AIR Software Quality Assurance Checklists

TABLE 1 Software Quality Evaluation Checklist

Reviewer _____ Project _____
 Date(s) of Review _____ Products Examined _____

		YES	NO	N/A	NOTES
1.	Has the Software Quality Assurance Plan (SQAP) been updated to make it relevant to the current status and activities of the project?				
2.	Is the SQAP being followed?				
3.	Is the project on schedule and are the SQA activities up-to-date with this schedule progress?				
4.	Is the project following corporate SQA standards and procedures?				
5.	Is the project following the additional standards and procedures required by the customer and application area?				
6.	Have regular and periodic reports about the projects SQA activities and accomplishments been submitted to the corporate SQA manager?				
7.	Are records of results of all SQA evaluations being kept in the project's SQA files?				
8.	According to the records in the SQA files, have corrective actions been processed in a timely manner?				
9.	Are all checklist negative answers corrected or noted in documented issues?				
10.	Have all internal reviews been held as scheduled and have records of these reviews been filed?				
11.	Have all reviews of support functions (Software Configuration Management, Software Development Library, etc.) been held? Are support functions operating effectively?				
12.	Are problem reports being completely and correctly filled out?				
13.	Are follow-up corrective actions being documented and implemented?				
14.	Are impacts/side effects of corrective changes being investigated and resolved (impact on other modules, user documentation, etc.)?				
15.	Are the causes of problems being traced to their root causes? Are the development and SQA processes being changed to eliminate these root causes?				
16.	Are the data for software metrics being recorded and evaluated?				

TABLE 1 Software Quality Evaluation Checklist (cont.)

Reviewer _____
Date(s) of Review _____
Project _____
Products Examined _____

	YES	NO	N/A	NOTES
17. Are the sign-off procedures prior to delivery being followed? Have any products been released or delivered without SQA approval or over the objections of SQA?				
18. Does the project SQA representative meet frequently with software development leaders to discuss plans, accomplishments, problems and concerns?				
19. Does the SQA representative leader report frequently to the project manager regarding status and problems of the software development process and products?				

TABLE 2 Software Configuration Management Review Checklist

Reviewer _____ Date(s) of Review _____

Project _____ Products Examined _____

	YES	NO	N/A	NOTES
1. Has the Software Configuration Control Board (SCCB) met regularly?				
2. Do SCCB members constitute all disciplines of the project organization?				
3. Is the Requirements Baseline (RBL) under formal Software Configuration Management (SCM) baseline control?				
4. Does the RBL have approval status?				
5. Are all updates and changes to the RBL approved and accounted for?				
6. Is the Functional Baseline (FBL) under formal SCM baseline control?				
7. Does the FBL have approval status?				
8. Are all updates and changes to the FBL approved and accounted for?				
9. Is the Allocated Baseline (ABL) under formal SCM baseline control?				
10. Does the ABL have approval status?				
11. Are all updates and changes to the ABL approved and accounted for?				
12. Is the Developmental Baseline (DBL) under formal SCM baseline control?				
13. Does the DBL have approval status?				
14. Are all updates and changes to the DBL approved and accounted for?				
15. Is the Product Baseline (PBL) under formal SCM baseline control?				
16. Does the PBL have approval status?				
17. Are all updates and changes to the PBL approved and accounted for?				
18. Is there an up-to-date list of all units in a software component?				
19. For all software units required to support a given test:				
a. Are all required units uniquely identified and uniquely name/numbered?				
b. Do all required units have updated program listings with unique identifiers?				
c. Are unit test results baselined?				

TABLE 2 Software Configuration Management Review Checklist (cont.)

Reviewer _____ Date(s) of Review _____ Project _____ Products Examined _____

		YES	NO	N/A	NOTES
20.	For the group of SQA approved units which have been integrated together into a build to support integration testing:				
a.	Has the build been baselined prior to test?				
b.	Are problems (encountered during the test) uniquely identified and accurately recorded against the current build?				
c.	Are test plans reviewed, approved, and baselined prior to test?				
d.	Are test procedures reviewed, approved, and baselined prior to test?				
21.	For formal, deliverable Product Baselines:				
a.	Are quality concerns satisfied for the deliverable version of the Functional Baseline Products?				
b.	Are quality concerns satisfied for the deliverable version of all baselined products?				
c.	Is all software baselined?				
d.	Are all version and revision identifiers unique and current?				
e.	Are all test plans and procedures updated and approved?				
f.	Are all test results certified by SQA as accurate findings of the tests?				
g.	Are all problem reports and changes approved and closed out?				
h.	Have all waivers or deviations been accepted by the customer?				
22.	Are there any out-of-date files in the SCM system?				
23.	Have any files been reserved for an unusually long period of time?				
24.	Has the Software Configuration Management Plan (SCMP) been kept up-to-date?				
25.	Are copyright statements included in each component?				
26.	Is corresponding source and executable code archived?				
27.	Is an up-to-date master list maintained of all archived media?				
28.	Is an up-to-date compilation list maintained for each software components?				
29.	Are all interface documents updated and approved?				

TABLE 3 Software Development Library (SDL) Review Checklist

Reviewer _____ Project _____
 Date(s) of Review _____ Products Examined _____

		YES	NO	N/A	NOTES
1.	Is there a recognized approved SDL Control Procedure in the Configuration Management Plan?				
2.	Does the SDL control procedure discuss how the library function coordinates with, augments, and complements (without being redundant) the function of CM?				
3.	Does the SDL control procedure define what program materials are to be entered into the SDL for this particular project?				
4.	Has SDL identification of library materials been carried out in compliance with CM practices?				
5.	Has SDL status accounting of library materials been carried out in compliance with CM practices?				
6.	Does the SDL differentiate and separate (in physical storage) software under development and software under formal baseline control?				
7.	Has the mechanism for placing all materials into the library (review, signature approval, and acceptance) been followed?				
8.	Is there a complete up-to-date status log of all materials contained in the library?				
9.	Is there a complete up-to-date status log of all approved changes to library materials?				
10.	Are all library materials (master tapes, discs, cards, listing documents, logs, etc.) protected from inadvertent or unauthorized use?				
11.	Are all library materials protected from inadvertent unauthorized change?				
12.	Are copies verified when made? (e.g., Is a comparator tool used for verifying the accuracy of software copies?)				
13.	Has the mechanism for releasing materials from the library (signature approval and release) been followed?				

TABLE 3 Software Development Library (SDL) Review Checklist (cont.)

Reviewer _____

Date(s) of Review _____

Project _____

Products Examined _____

		YES	NO	N/A	NOTES
14.	Are library logs maintained to assure a change against baselined software results in corresponding changes to affected documents and specifications?				
15.	Are library logs maintained to assure a change against a baselined document results in corresponding changes to software and other affected documents and specifications?				

TABLE 4 Corrective Action System Evaluation Checklist

Reviewer _____ Project _____
 Date(s) of Review _____ Products Examined _____

	YES	NO	N/A	NOTES
1. Are problem reporting procedures clearly documented and usable?				
2. Are problem reporting procedures distributed to all staff who need to use them?				
3. Is there an analysis procedure and is this procedure adequately staffed?				
4. Is there a documented problem report categorization and prioritization procedure?				
5. Is there a procedure for analysis of problem trends? Is this procedure followed?				
6. Is there a documented procedure for submitting and following up on recommended corrective actions? Does this procedure include proper authorization of actions? Is this procedure followed?				
7. Is there a thorough and complete documentation and record-keeping procedure for problems and corrective actions?				
8. Have evaluations been conducted of the effectiveness of corrective actions after they have been taken?				
9. Is there a documented procedure, with proper authorization, for closing out completed corrective actions?				
10. Have plans been made to facilitate customer visibility into the corrective action system?				
11. Are there procedures for identifying problems of traceability between the requirements and preliminary design? Preliminary design and detailed design? Detailed design and code? Are these procedures followed?				

TABLE 5 Hardware - Software Allocation Checklist

Reviewer _____ Project _____
 Date(s) of Review _____ Products Examined _____

		YES	NO	N/A	NOTES
1.	Have interface specifications been agreed upon by those departments/organizations with software and hardware responsibilities?				
2.	Is new special-to-type hardware a feature of the project?				
3.	Is the special hardware the processor and /or its memory?				
4.	Are 'power-up' and power-fail' procedures specified and agreed upon?				
5.	Have sufficient software tests, using simulation or modeling techniques where required, been devised at various levels to exercise the software throughout its progressive development?				
6.	Are special hardware facilities required to simulate eventual hardware and allow software proving?				
7.	Where the new hardware is functionally similar to previous items, has consideration been given to common input/output codes?				
8.	Are system start-up delays required? If so, are they hardware or software implementation and can they be over-ridden: (a) in a functioning system? (b) during test to permit fault finding?				
9.	Are occasional input/output errors permitted, detectable, logged, recoverable?				
10.	Will it be possible to integrate and test hardware and software sub-assemblies prior to full system integration?				

TABLE 5 Hardware - Software Allocation Checklist (cont.)

Reviewer _____ Date(s) of Review _____ Project _____ Products Examined _____

		YES	NO	N/A	NOTES
11.	Are separate integration specifications required, available, and agreed upon?				
12.	Are responsibilities for monitoring and documenting integration defined?				
13.	Is hardware or software data buffering employed across the interfaces? If so, is overflow action specified and agreed upon?				
14.	Do standard driver routines exist for this hardware?				
15.	If special drivers are required are they to exist at the operating system level and conform to its rules?				
16.	How far will software integrity be established against the full speed range of hardware timing possibilities?				
17.	Is special software required to verify hardware and interface design?				
18.	Are special hardware checking routines required: (a) as part of the 'normal' system software? (b) for periodic maintenance or servicing?				
19.	Are the special hardware checking routines written, tested and agreed by the responsible hardware and software leaders?				
20.	If system software detects a hardware malfunction, can it: (a) use other hardware in a reconfigured mode? (b) notify or record the malfunction?				
21.	Can hardware faults be simulated to check (a) and (b) above?				
22.	Is special test equipment required and available for interface debugging?				

TABLE 6 Software Requirements Document Review Checklist

Reviewer _____		Project _____	
Date(s) of Review _____		Products Examined _____	

	YES	NO	N/A	NOTES
1. Is the system's functional flow clearly and completely described?				
2. Has each decision, selection and computational function that the software must perform been clearly defined?				
3. Is there a complete definition of software functions requiring human reactions and operator response?				
4. Is there a description of the performance and capacities required of each function?				
5. Does each software function that is specified trace to one or more system requirements?				
6. Is there a description of the inputs and outputs of each function?				
7. Can the required performance and capabilities be achieved?				
8. Are all common functions identified and appropriate linkage mechanisms enabling their common use defined?				
9. Is a dictionary for all data elements provided and is it complete?				
10. Do the input descriptions match the output descriptions for inter-functional communications?				
11. Are timing requirements given?				
12. Are memory requirements given?				
13. Are the timing and memory limits compatible with hardware constraints and with system timing and capacity budgets?				
14. Are all limits and restrictions on software performance defined?				
15. Is there a description of the executive system requirements?				
16. Are the support software provisions and limitations given?				
17. Is the method for demonstrating compliance with requirements defined and are all limitations of these methods explained?				
18. Are standards and naming conventions established for the project followed?				
19. Are all functions testable?				
20. Are required software quality attributes (e.g., reliability, portability, reusability, maintainability, etc.) defined?				

TABLE 7 Software Design Document Review Checklist

Reviewer _____ Project _____
 Date(s) of Review _____ Products Examined _____

		YES	NO	N/A	NOTES
1.	Is the structure of the software compatible with the functional requirements and with sound design practices?				
2.	Is there a complete listing of all symbols and definitions?				
3.	Is there a complete listing of all symbol meanings and parameter values?				
4.	Does each unit description include a statement of the functions to be performed by that unit?				
5.	Is there a unit or combination of units to perform every required function?				
6.	Does functional media show the relationships, both data and control, between all units?				
7.	Are all conditions of system status that call for a unit response described?				
8.	Is the flow of input and computed output shown?				
9.	Do the input descriptions match the output descriptions of the unit generating that output?				
10.	Are the specifications of all interfaces with the software executive shown?				
11.	Are the algorithms or equations to be evaluated by each unit completely and correctly presented?				
12.	Are the entry and exit points of each unit listed?				
13.	Are all units that may be called by this unit listed?				
14.	Are all the units that may call this unit listed?				
15.	Has the logic within each unit been given and the decision criteria been defined, whether system status, operational mode, or computed?				
16.	Is there a description of software error conditions, error exits, and recovery procedures?				
17.	Is the design of sufficient detail to allow coding?				

TABLE 7 Software Design Document Review Checklist (cont.)

Reviewer _____ Date(s) of Review _____

Project _____ Products Examined _____

		YES	NO	N/A	NOTES
18.	Can the units be tested as designed?				
19.	Is the unit's functional flow clearly described?				
20.	Are all common functions identified and appropriate linkage mechanisms enabling the sharing of these functions defined?				
21.	Are specified buffer sizes adequate?				
22.	Are there adequate margins for minimum/maximum data values?				
23.	Have design conventions and standards been followed?				
24.	Is the functional role of each table and block clearly defined?				
25.	Do these data objects fulfill these stated functions?				
26.	Is the scope of each table and block clearly defined?				
27.	Is each table's and block's structure clearly described, giving a pictorial representation, and description of each field's name, size, location, contents, and level?				
28.	Is there a list of all units reading from, or writing into, each table or block?				
29.	Is each table organized in a logical and efficient manner, consistent with its use by various units at different levels in the program hierarchy?				
30.	Is there a correct description of the input from each unit including its type, source, format, units, accuracy, and quantity?				
31.	Is there a correct description of the output from each unit, including its type, destination, format, units, accuracy, and quantity?				
32.	Is undesirable underflow or overflow that is undetected avoided?				
33.	Are mathematical algorithms properly implemented?				
34.	Are interrupts properly handled?				
35.	Are all referenced data items declared?				
36.	Is the use of each data item valid (based on its type designation)?				
37.	Are all declared data items actually used in this unit?				

TABLE 7 Software Design Document Review Checklist (cont.)

Reviewer _____ Date(s) of Review _____ Project _____ Products Examined _____

		YES	NO	N/A	NOTES
38.	Are the data items used only after they have been assigned values?				
39.	Do all of the unit's possible control paths terminate? (i.e., "no endless loops")?				
40.	Are all the calls made by this unit shown in the calling hierarchy?				
41.	Do all calls made by this unit follow the calling conventions of the programming guidelines?				
42.	Do all names defined in this unit follow the conventions of the programming guidelines?				
43.	Does the unit's control logic flow from top to bottom?				
44.	Does this unit have a single entrance and a single exit?				
45.	Does this unit determine the values of all inputs passed to units it calls?				
46.	Does this unit use all outputs returned by units it calls?				
47.	Does this unit avoid the use of function or control codes to determine the function or flow of control of units it calls?				
48.	Does this unit avoid using external data items that are used by other units in a manner that violates programming standards?				

TABLE 8 Software Code Review Checklist

Reviewer _____		Project _____	
Date(s) of Review _____		Products Examined _____	

	YES	NO	N/A	NOTES
1. Is unexecutable code avoided?				
2. Are all declared data items actually used?				
3. Is redundant code avoided?				
4. Are endless loops avoided?				
5. Are references to undefined data items (before assignment) avoided?				
6. Is argument passing compatible?				
7. Do all branches point to correct locations?				
8. Are all alternatives of conditional execution correct?				
9. Is initialization for all program states correct?				
10. Is common storage protected?				
11. Are constants defined with the correct values?				
12. Are mathematical algorithms properly implemented?				
13. Are buffer sizes adequate?				
14. Are margins for minimum/maximum data values adequate?				
15. Is the program logic correct?				
16. Are comments clear, concise, consistent, and sufficient and follow the standards?				
17. Have coding standards and conventions been followed?				

TABLE 9 Software Test Plan Review Checklist

Reviewer _____ Date(s) of Review _____ Project _____ Products Examined _____

		YES	NO	N/A	NOTES
1.	Is the location and schedule for the testing given?				
2.	Are the responsibilities of each of the test personnel clearly delineated?				
3.	Are the objectives of the test effort described and are they compatible with program objectives?				
4.	Will each requirement be demonstrated as fulfilled by means of the test effort?				
5.	Does all testing to be done aid in demonstrating compliance with the defined requirements?				
6.	Does the proposed test effort correspond to the structure of both the requirements and the software?				
7.	Are all applicable documents, user's manuals and programmer guides referenced?				
8.	Are methods given for evaluating whether units correspond to individual unit specifications?				
9.	Are limitations of test methods and facilities described, and are adequate steps taken to minimize the effort of those limitations?				
10.	Is provision made for the review and verification of test input data?				
11.	Are the general configurations of the software and test environment described?				
12.	Is the conduct of the tests described?				
13.	Are there procedures for the configuration control of test tools and facilities?				
14.	Are there procedures for the configuration control of software elements under test?				
15.	Is adequate provision made for regression testing?				
16.	Is the content of the test reports described and will the planned tests contribute the information needed to complete the test reports?				

TABLE 9 Software Test Plan Review Checklist (cont.)

Reviewer _____ Date(s) of Review _____ Project _____ Products Examined _____

		YES	NO	N/A	NOTES
17.	Is the anticipated test coverage given and is this coverage adequate?				
18.	Are requirements for special data reduction and analysis efforts described?				
19.	Does the test effort make adequate provision for the control and incorporation of changes to the specification, design, or code that may occur during the test effort?				

TABLE 10 Software Test Procedure Review Checklist

Reviewer _____ Project _____
 Date(s) of Review _____ Products Examined _____

		YES	NO	N/A	NOTES
1.	Is each test specified in the test specifications covered?				
2.	Does each procedure completely describe instructions for conducting the test?				
3.	Does the test suite complement and extend the unit test suites? Is the correct test environment being considered for each type of test?				
4.	Are sufficient test sets used in each type of testing?				
5.	Is there criteria for determining which tests will be run?				
6.	Are all tests and supporting software under configuration control?				
7.	Is all software to be tested under configuration control?				
8.	Are testing schedules established and followed?				
9.	Is there tracking of daily, weekly, and cumulative test results?				
10.	Are backup tapes maintained for all testing baselines?				
11.	Are there sufficient procedures for regression testing?				

TABLE 11 Software Test Conduct Review Checklist

Reviewer _____ Project _____
 Date(s) of Review _____ Products Examined _____

		YES	NO	N/A	NOTES
1.	Was the hardware and software configured as specified in the test procedure?				
2.	Was the global test environment set up as specified in the test procedure?				
3.	Were the test input data values those specified in the test procedures?				
4.	Was each test conducted as specified in the test procedure?				
5.	Were all tests performed in the sequence specified in the test plan?				
6.	Was each test recorded as specified in the test procedure?				
7.	Do test results correspond to expected test results?				
8.	Does the application of the test criteria indicate a correct result?				
9.	Can differences between the expected results and the observed results be reconciled?				
10.	Is additional testing required to isolate problems and trace them to their causes?				